Chapter 6

Motor Control

Motor Control

Roland Luder

1. INTRODUCTION

As already mentioned in the chapter AUTOMATION SYSTEMS, Motor Control is a very important part of cement plant control. It is the part which switches drives and valves etc. ON and OFF.

These switching operations have to be performed under normal as well as under special operating conditions of the process. Motor control further includes monitoring and alarming of these conditions. Physically, a motor control system consists of

- an operating panel as interface to the operator (with push buttons, lamps, mouse, displays)
- a logic controller which performs the logical interlocking of all information (with relays, electronic or programmable controllers)
- a plant/process interface with sensors and command elements (switches, contactors, valves)
- cabling, which interconnects the different components of the system.

A motor control system for a modern cement production line has to control approx. 1000 motors and a few hundred valves, heaters, acoustic and optical devices. The correct and efficient command of all these items requires a systematic planning of the system and a careful selection of the components.

The tendency in systems design is to integrate motor control and analogue control functions into the same physical unit. This set-up has the advantage that only one type of hardware is required and that interconnections between the binary and the analogue systems can be programmed and do not have to be wired.

2. PROGRAMMABLE CONTROLLERS (PLC)

2.1 Introduction, History

The logic controller is the brain of the motor control system, i.e. the decision-making part. It takes care of the safe, convenient sequential starting, running and stopping of single motors or of whole groups of machines.

There are basically three different techniques to build-up an electrical logic controller:

2.1.1 Relay Technology

A relay uses the electro-mechanical equipment which uses the electromagnetic force of an electrical coil to open or close electrical contacts. The internal wiring of these coils and contacts determines the specific function of the set-up.

Relays control has been the only technology until approx. 1960.

Today relays are still used for interfacing (power amplifying) purposes and for very small control systems.

2.1.2 Electronic Card System

This technology was developed in Europe and was applied approx. until 1975. It consists of electronic components, pre-assembled on printed circuit boards and performing specific logic functions. The function of the total system is determined by the wiring between the different logic cards.

Today these systems are hardly anymore installed.

2.1.3 Programmable Controllers

This technology was introduced in the market approx. 1970, and is the today's dominating technology. The tendency is to replace relays by programmable controllers even in very small applications. The programmable controller uses basically the same idea as a computer does: it uses a memory to store information and it uses this information to execute step by step a procedure which is determined by this information (program; software).

The advantages of the programmable controllers as opposed to other technologies are:

- No mechanical wear
- no rewiring when modifications are required
- easier planning (hardware / software can be planned in parallel)
- higher level of automation is possible
- integrated solutions with analogue control and data acquisition are possible.

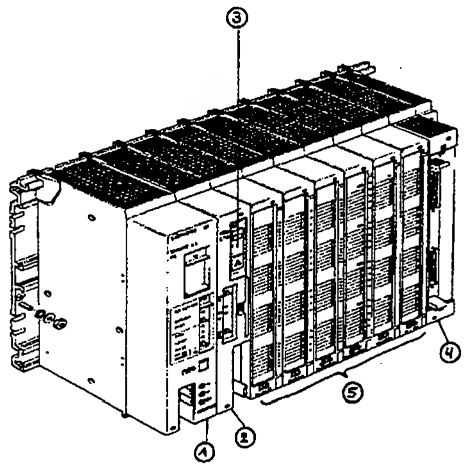
2.2 Hardware of Programmable Controllers

A programmable controller is basically a 'BLACK BOX' with INPUTS and OUTPUTS and a connection to a PROGRAMMING UNIT.

With the programming unit, the user determines how the different inputs and outputs must be logically correlated and sequenced to perform the desired control task. The programming nowadays uses a PC to display, to enter or to modify the program as well as an additional storage devices to safeguard it.

The 'BLACK BOX' is composed on one or several chassis or racks which basically contain the following elements (see Fig. 2.2.1)

Figure 2.2.1 Programmable Controller Hardware Configuration



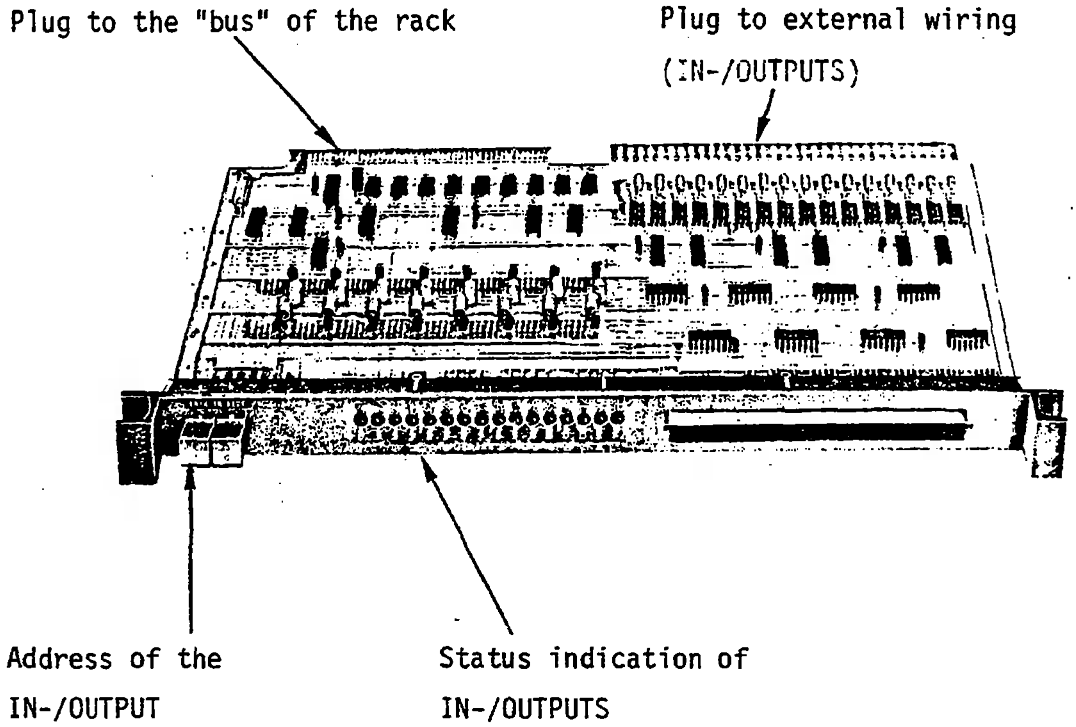
1) **Power supply**
   To provide the internal stabilised control voltages of the programmable controller
2) **Processor**
   To perform the actual logic / timing and internal control functions
3) **Memory**
   To store the program and data
4) **Communication port**
   To communicate with external devices (as programming panel, computer, other programmable controllers) or with remote input/output devices.
5) **Input/Output (I/O) Devices**
   To transform the signal coming in (INPUT) from outside (e.g. from a level switch) or going out (OUTPUT) to the outside (e.g. to lamps, to the MCC). Today most programmable controllers handle as well analogue I/O signals.

Physically all these elements are generally grouped on one or several cards of the draw-out type. The cards are internally interconnected via a system bus which performs a fast exchange of the necessary information between the different cards.

This arrangement provides great flexibility in hardware planning: if additional memory space is required, for examples, it is generally sufficient to plug in an additional memory card. The draw-out technique, of course, also simplifies troubleshooting of the hardware (defective card out - new card in - restart). It must be noted, however, that once a system is correctly set-up, there are generally no more hardware failures.

Figure 2.2.2 shows an example of an input card of the draw-out type which contains 16 inputs. Output cards may be arranged in a similar way.

Figure 2.2.2: INPUT (OUTPUT) Card of a Programmable Controller



The size of a programmable controller can be expressed by different figures. The most important figure is the number of I/O which a programmable controller can handle. The smallest units start at approx. 15 I/O, the biggest go up to 8000 I/O.

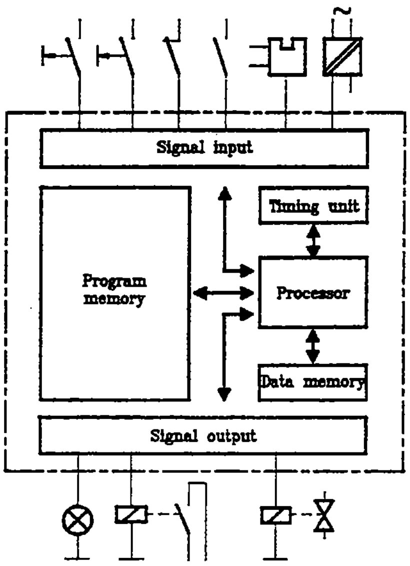
The maximum memory size is another key figure which, however, generally goes in parallel with the number of I/Os. Normal sizes range from approx. 0.5 K up to 1000 K (1 K = approx. 1000 program steps).

The cycle time (see 'software') generally goes in parallel with the size of the memory used and is generally expressed in ms/1 K. It ranges from approx. 0.5 to 50 ms/1 K.

2.3 Structure of a PLC

The diagram below shows the structure of a PLC. The main functional elements of a programmable logic controller are the control unit with one, or sometimes several microprocessors and the corresponding memories for data (timers, counters, markers, etc.) and programs (programmable memories).

Block diagram of programmable controller



The program memory, processor, counter, data memory and the input/output units are interconnected. Here connection via bus has become standard practice. By means of this bus the data are exchanged between data memory, processor and program memory.

2.3.1 Differences between PLC and a Computer

What are the main differences? In a PLC so-called bit processing is used. This is special processing method which processes only one bit. In contrast, the computer always uses word processors, i.e. single bits can only be addressed by programming.

A PLC functions in much the same way as a computer, but with the following main differences.

- User's programs are executed cyclically
- A PLC needs a very simple operating system
- The information is processed a bit at a time (facilities for word processing are available)
- A PLC is a real-time system, i.e. the results of operations are obtained within a short, clearly defined time
- The set of command is especially intended for control requirements and is therefore limited in its scope

- The hardware is designed for rough industrial conditions (temperatures between -10 and +60°C)
- Programming is simple, can be understood by electricians and is easy to learn
- Addressing of input and output cards is transparent
- To program a PLC a special programming unit has to be connected to it.

2.4 Software of Programmable Controllers

In the chapter 'Hardware of Programmable Controllers' we have seen that a programmable controller can be represented as a 'BLACK BOX' which contains inputs and outputs.

The different inputs and outputs have now to be logically and sequentially interconnected to perform the desired control task. To represent this "logical and sequential interconnection", special languages have been elaborated. Unfortunately, these languages or graphic presentations are not standardised. Main differences can be found between European and American presentations but even within one language, practically every suppliers uses his own 'slang'.

For a long time there was a certain market tendency toward the ladder diagram, influenced by the American market where generally only this language is used.

The ladder diagram is based on the representation which was used for relay systems. It can, therefore, easily be learned by people who worked with those systems but it does not well represent the new thinking in inputs/outputs.

Nowadays a standard is on the market named IEC1131.

Any logic can actually be represented with only 4 different instructions:

Logic AND, logic OR, logic NOT and TIME instruction. In order to make programming easier, all programmable controllers use additional logic instructions which are composed of specific often-used combinations of above four elements.

When a programmable controller is equipped to accept analogue inputs, additional instructions for arithmetic operations and file handling are available.

For bigger applications (as e.g. in the cement industry) a structured programming making use of 'macros' or subroutines should be applied. The same applies when analogue control capabilities are included in the programmable controller.

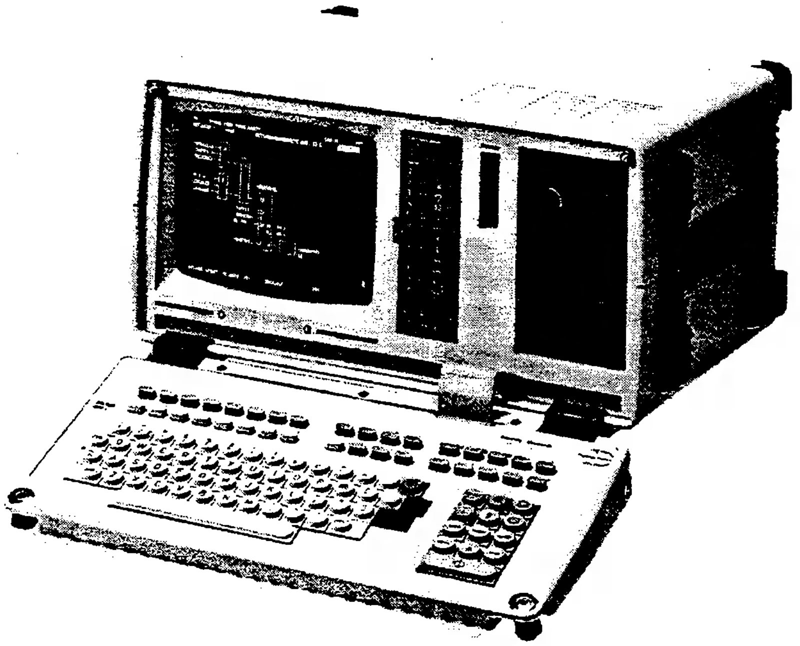
The actual program is now composed of a series of program steps, every step uses a combination of instructions or 'macros' to define how the different inputs have to be linked with the outputs.

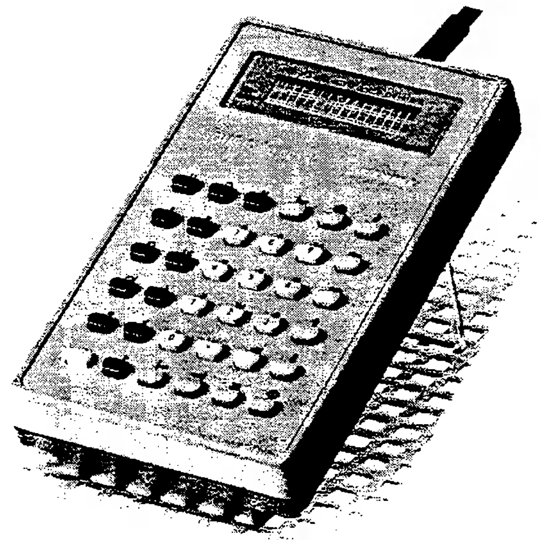
It is important to mention that the program is executed step by step. This means that one instruction after the other is read, interpreted and executed.

At the end, the program automatically restarts at the beginning. The total time which a program needs to come once from the beginning to the end is called the system cycle time. This time is generally very short (approx. 1...300 ms). For an external observer of the system it, therefore, behaves as if everything (all commands) would be immediately executed - all at the same time. In reality, as explained, only one instruction is executed at one time.

2.5 Programming a PLC

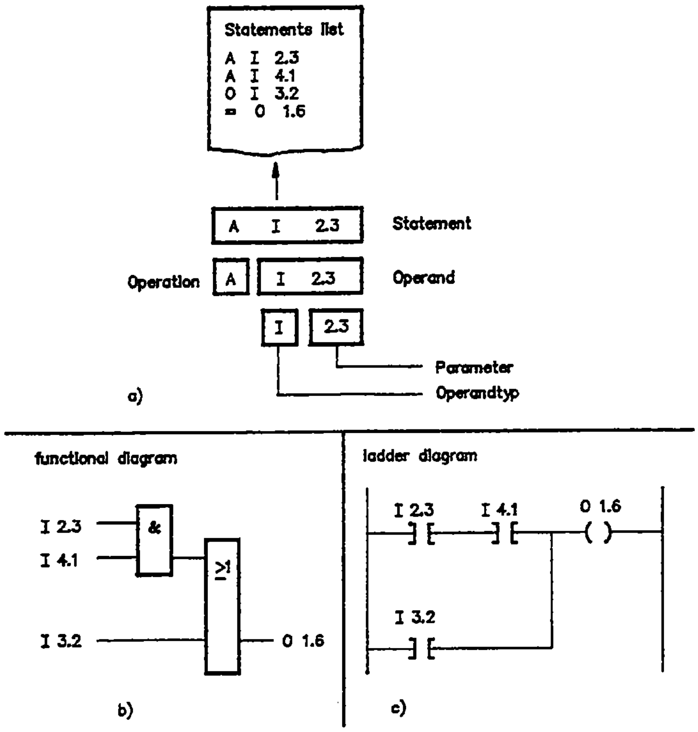
In order to program a PLC a special programming unit is required. The pictures below show different versions of programming units.





Programming units are very efficient, they are PC's of special industrial design, equipped with large-area LCD or with a monitor. They can translate the functions entered by the user in a higher-level programming language specifically intended for control tasks, e.g. as a list of instructions (Fig. a), as functional diagram or contact diagram, direct into the machine code of the control units. They can also translate from the machine code back into the higher-level representation.

Figure (a) Programming languages (Siemens)



A control task can be expressed in different 'languages'; the list of instructions (a) can also be translated by simple programming units into a machine language understood by the control system. The functional diagram (b) which is easier to understand by engineers and the ladder diagram (c) which is very popular in USA, impose more exacting demands on the programming unit.

At all events such units can be employed on-line, i.e. connected direct with the controller. The programs to be entered in this case are usually entered direct in the programming language of the controller, or for display purposes are read out of a memory. In on-line operation it is also possible to perform test functions, fault location and program correction.

A further important function of the programming unit is the program documentation. Special documentation software converts the programs into a form readily understood by the user (auxiliary text, I/O description, comments, etc.). A printer connected to the system is able to print out the documented programs.

2.6 Criteria governing the Choice of PLC

Basically, any PLC can be used for control tasks in the cement industry, provided it satisfies to following requirements.

2.6.1 General

- The system is well represented and generally known in the country
- Spare parts are guaranteed obtainable at least for 10 years
- The system must be capable for expansions in order to integrate future adaptations
- The dimensions of equipment permit the replacement of existing facilities

2.6.2 Central Unit (CPU)

- When the system is extended to full capacity, the cycle time should not exceed 150 ms
- Adapted memory capacity, so that there is no shortage of storage capacity when the system is fully expanded.
- On-line programming of modifications while the process is in progress
- Reasonable set of instructions containing the following:
  arithmetic with variable decimal point, PID algorithms and functions specified by the user, modules (e.g. motor module).
- Floating point arithmetics

2.6.3 Communication

- Between PLC's with bus
- Standardised interface with simple protocol with a main-frame computer (e.g. RS 232, RS 422, RS 485)
- Standardised interface with subcontrol systems (field bus, profibus, RS 232)

2.6.4 Inputs/Outputs

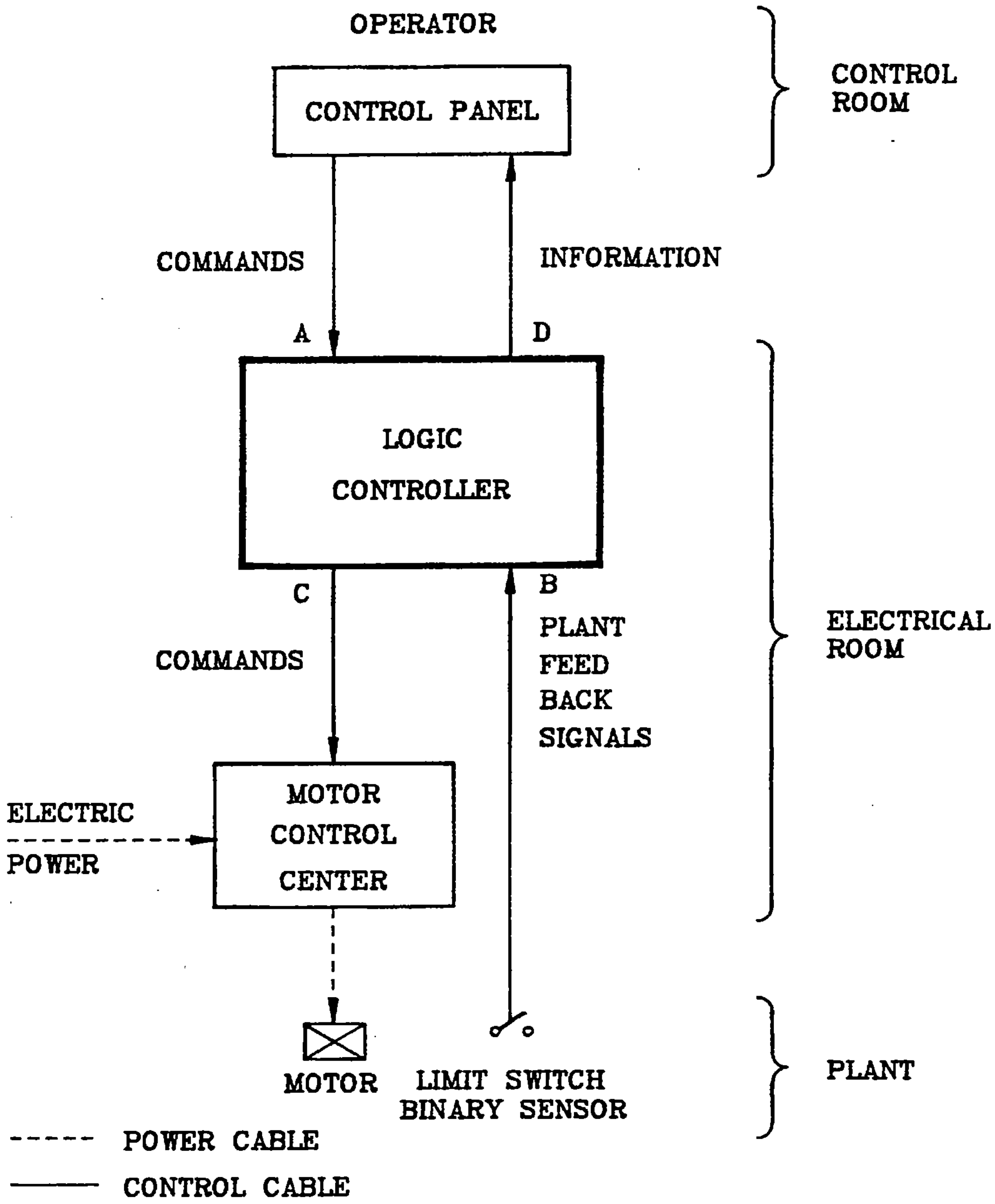
- Decentralised peripherals connected with the CPU by a serial line (possibly an optical link)
- 24 V or 48 V DC single-ended inputs/outputs with common ground
- 4-20 mA analogue single-ended inputs/outputs with common ground
- Capable of extension up to 2000 inputs and outputs

2.6.5 Programming and Documentation

- Off-line programming and documentation with standard PC's
- Remote line connection of programming units
- Graphic representation, preferably by functional diagram
- Symbolic programming of addresses with at least 15 freely chosen characters
- Commands and operating instructions in the local language

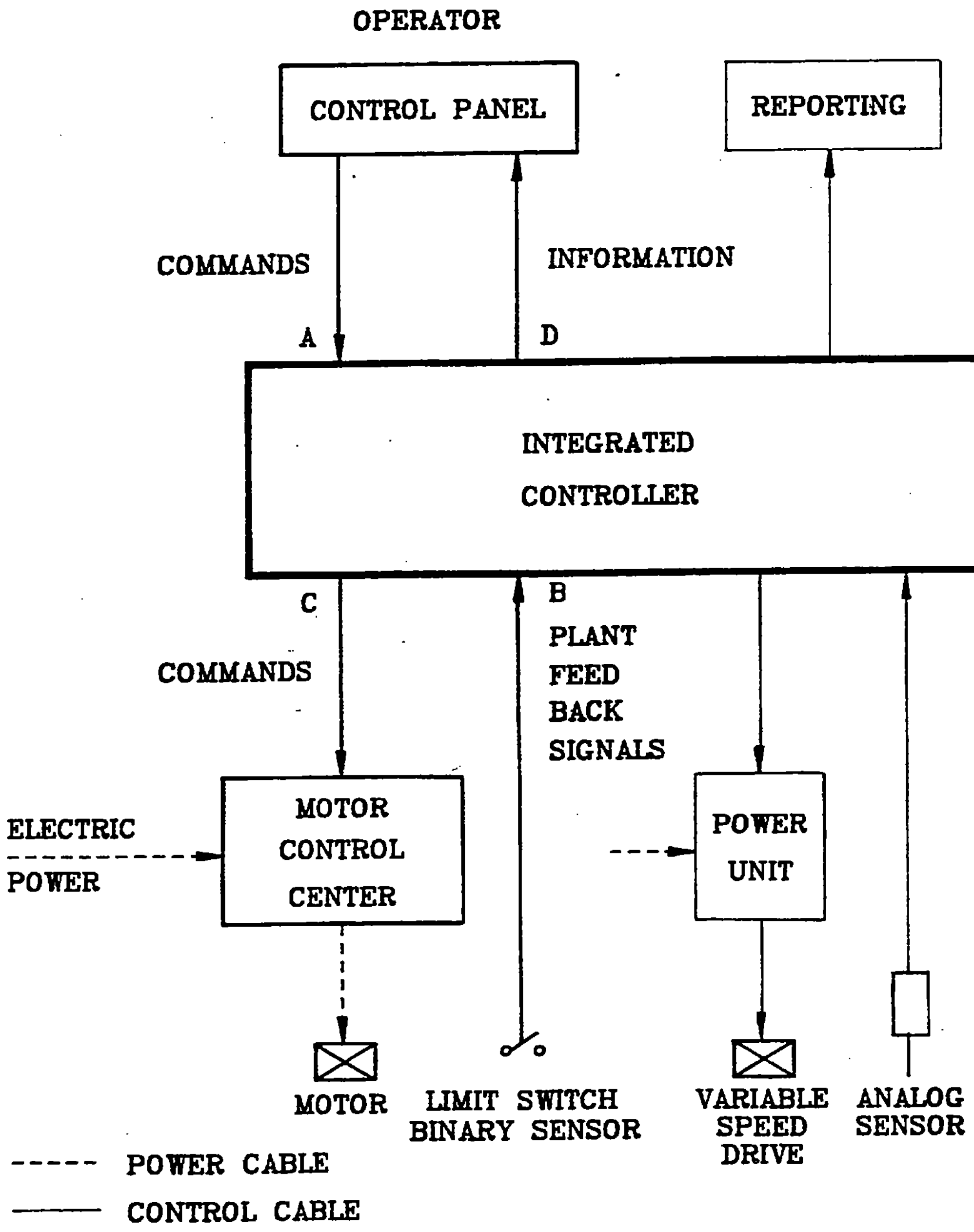
Motor control - Logic controller

Task: Sequence start and stop of motors, interlocking of motors, machine protection, alarming

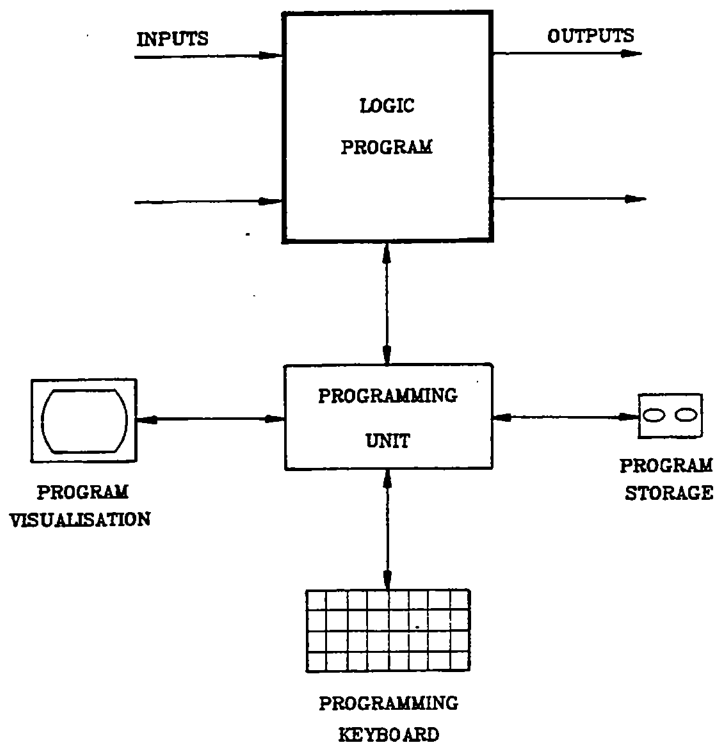


Motor control - integrated

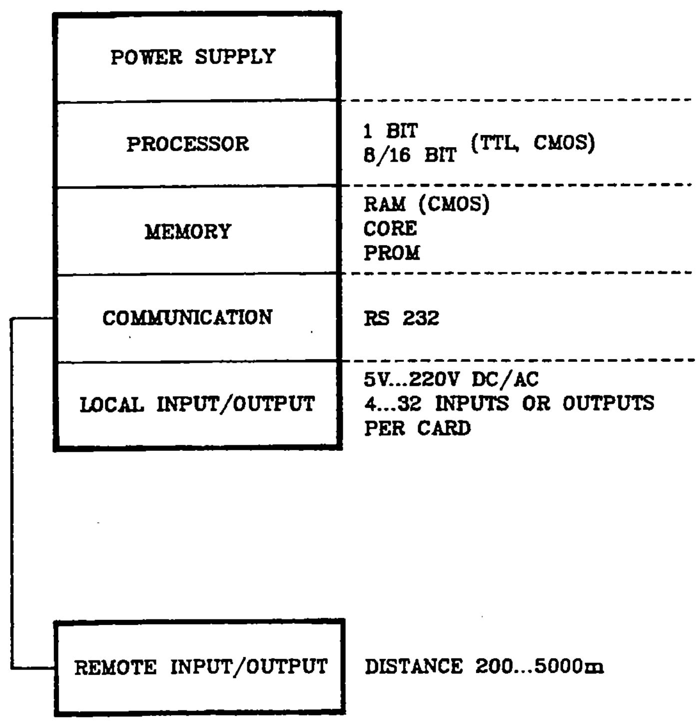
Task: Sequence start and stop of motors, interlocking of motors,machine protection, alarming, measuring, analog control, reporting



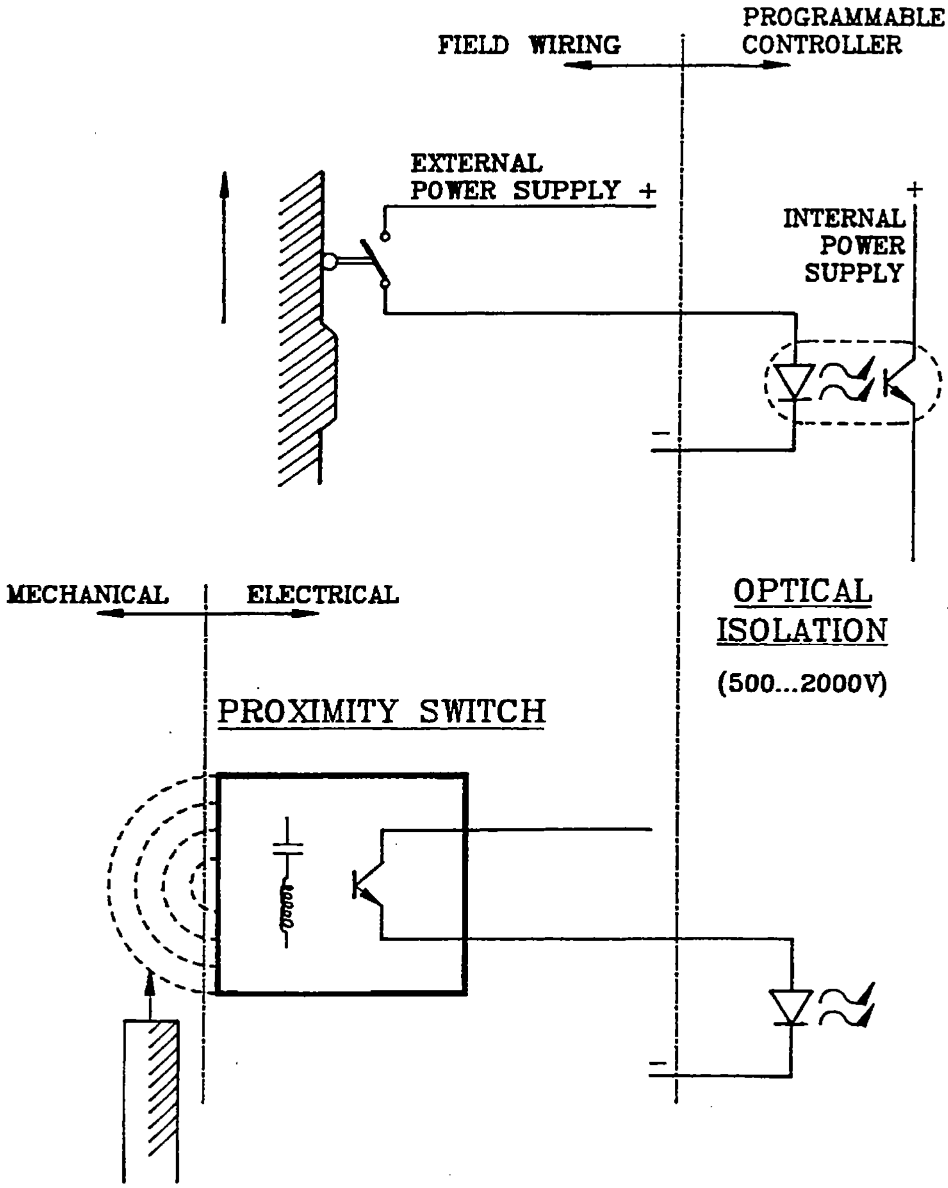
Motor control - Programmable controller: System configuration



Motor control - Programmable controller: Hardware configuration

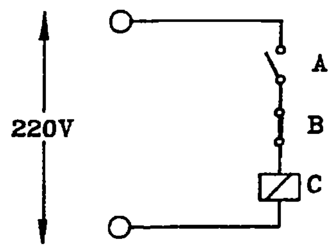


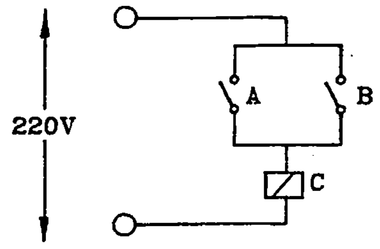
Motor Control - Programmable controller: Isolation of signals



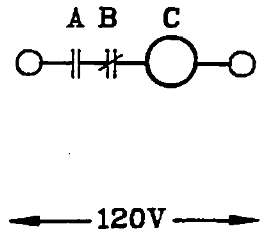
Motor Control - Programmable controller: Language, presentation

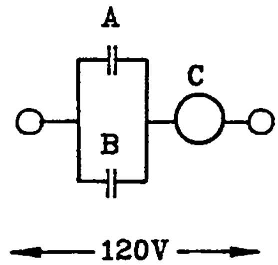
1. Relay European: AND NOT / OR



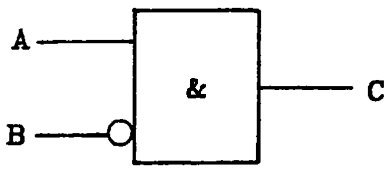


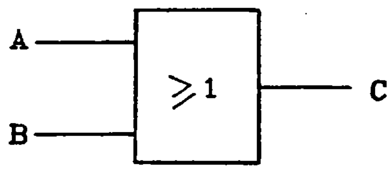
2. Relay American (Ladderdiagram): AND NOT / OR





3. Logic card European: AND NOT / OR

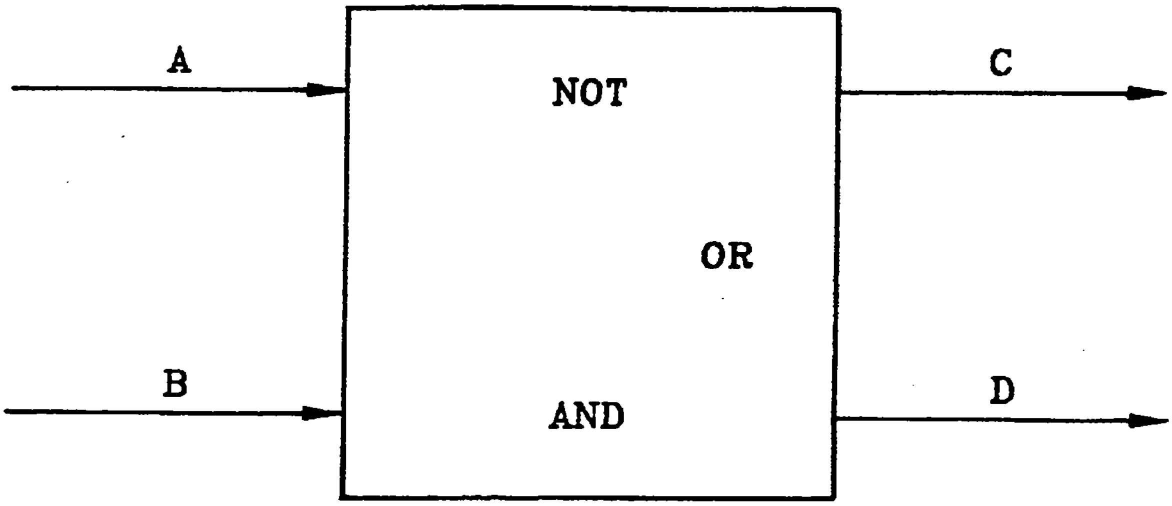




4. Programmable logic controller: Boolean language, (or logic, or ladder)

A AND NOT B = C **A OR B = C**

Motor Control - Programmable controller: Language, Example



Example:

A: Operator command: Turn motor left

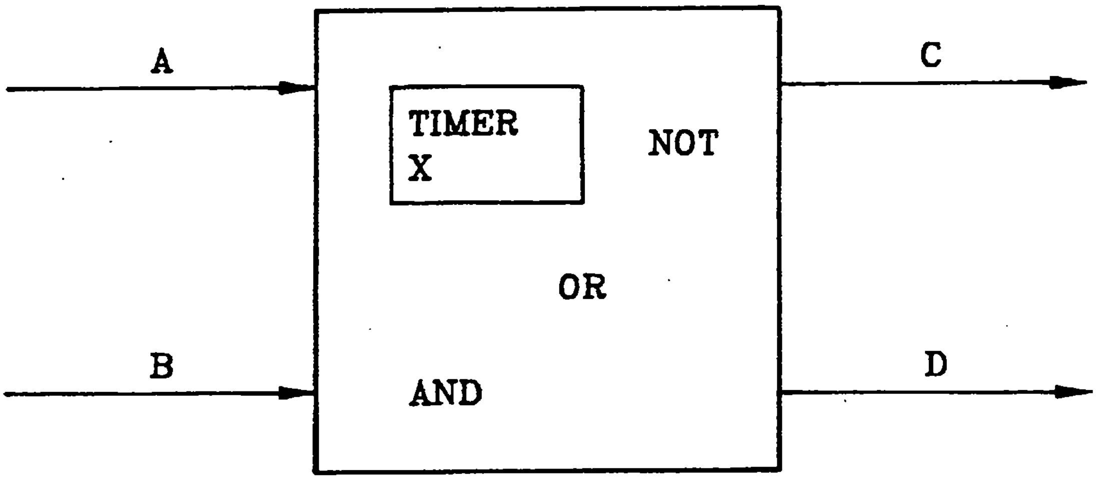
B: Plant feedback: Limit switch left reached

C: Control system command: Turn motor left

D: Operator information lamp: Device in position left

Program	**Short**
1. Input A	
2. AND NOT Input B	
3. Equals Output C	A AND NOT B = C
4. Input B	
5. Equals Output D	B = D

Motor Control - Programmable controller: Language, timer



<u>Example Timer:</u>

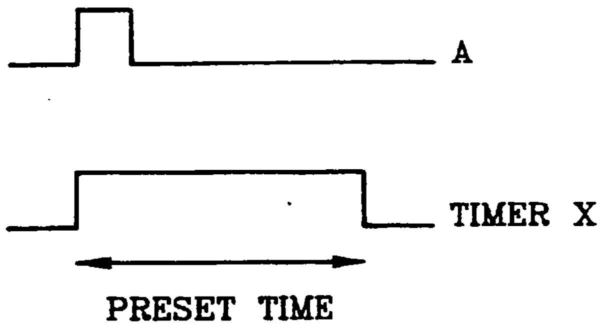
A: Operator command: Turn motor left

B: Plant feedback: Limit switch left reached

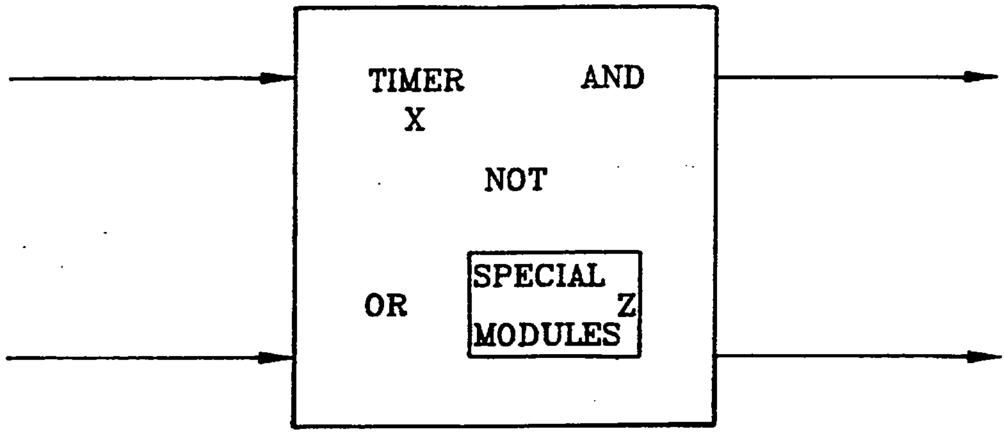
C: Control system command: Turn motor left

<u>Program</u>

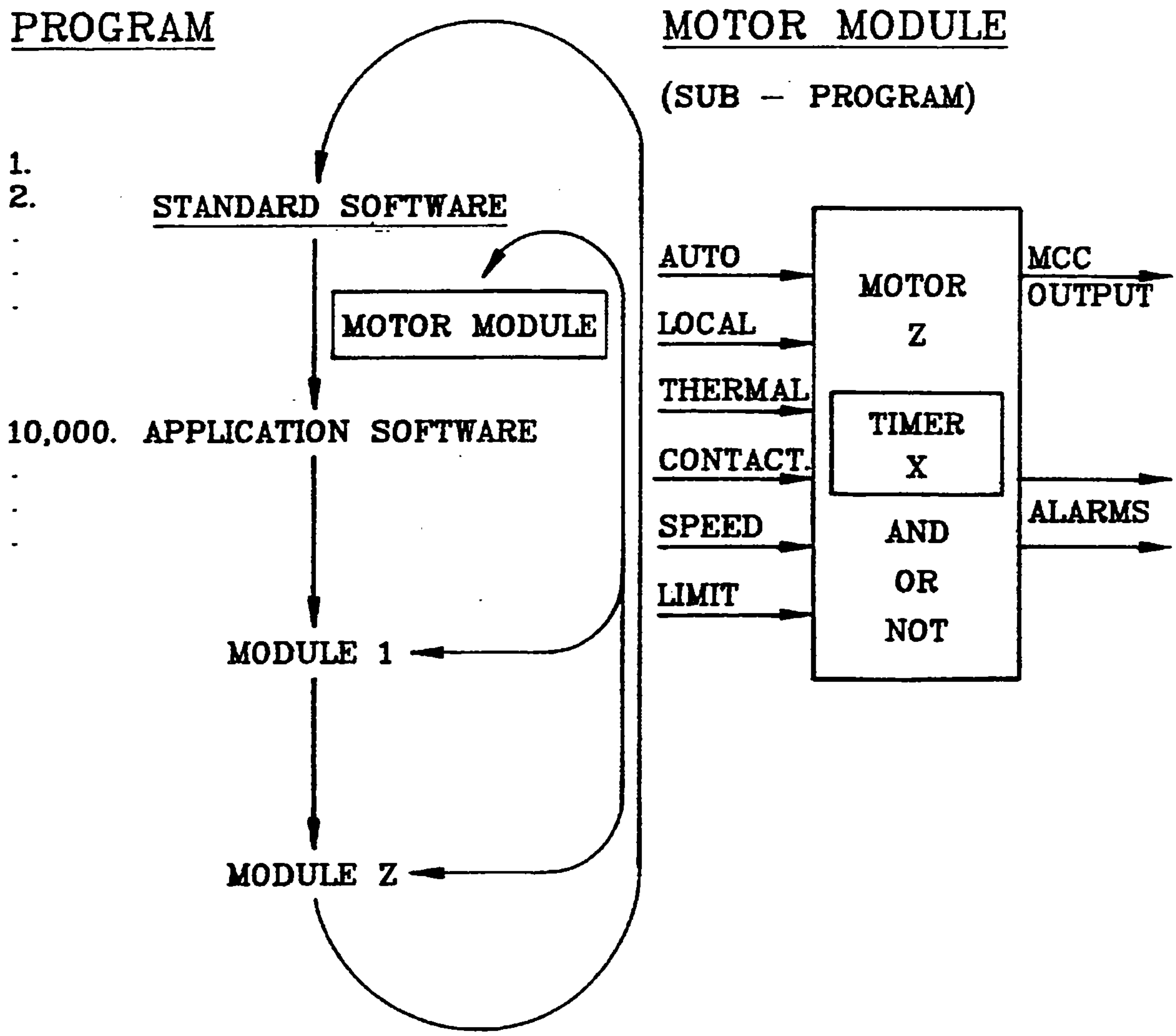
1. Timer X = 30 seconds
2. Input A
    3. Equals timer X
4. Timer X
5. AND NOT input B
    6. Equals output C



Motor control - Programmable controller: Language, structured programming



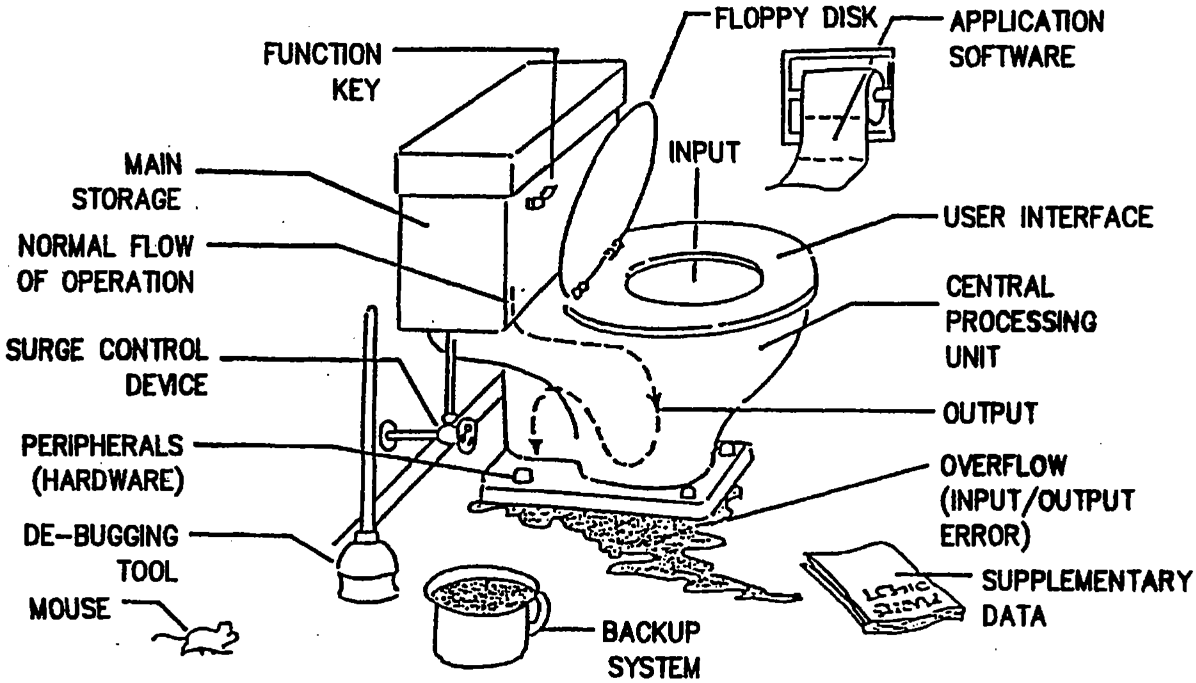
EXAMPLE MOTOR MODULE:



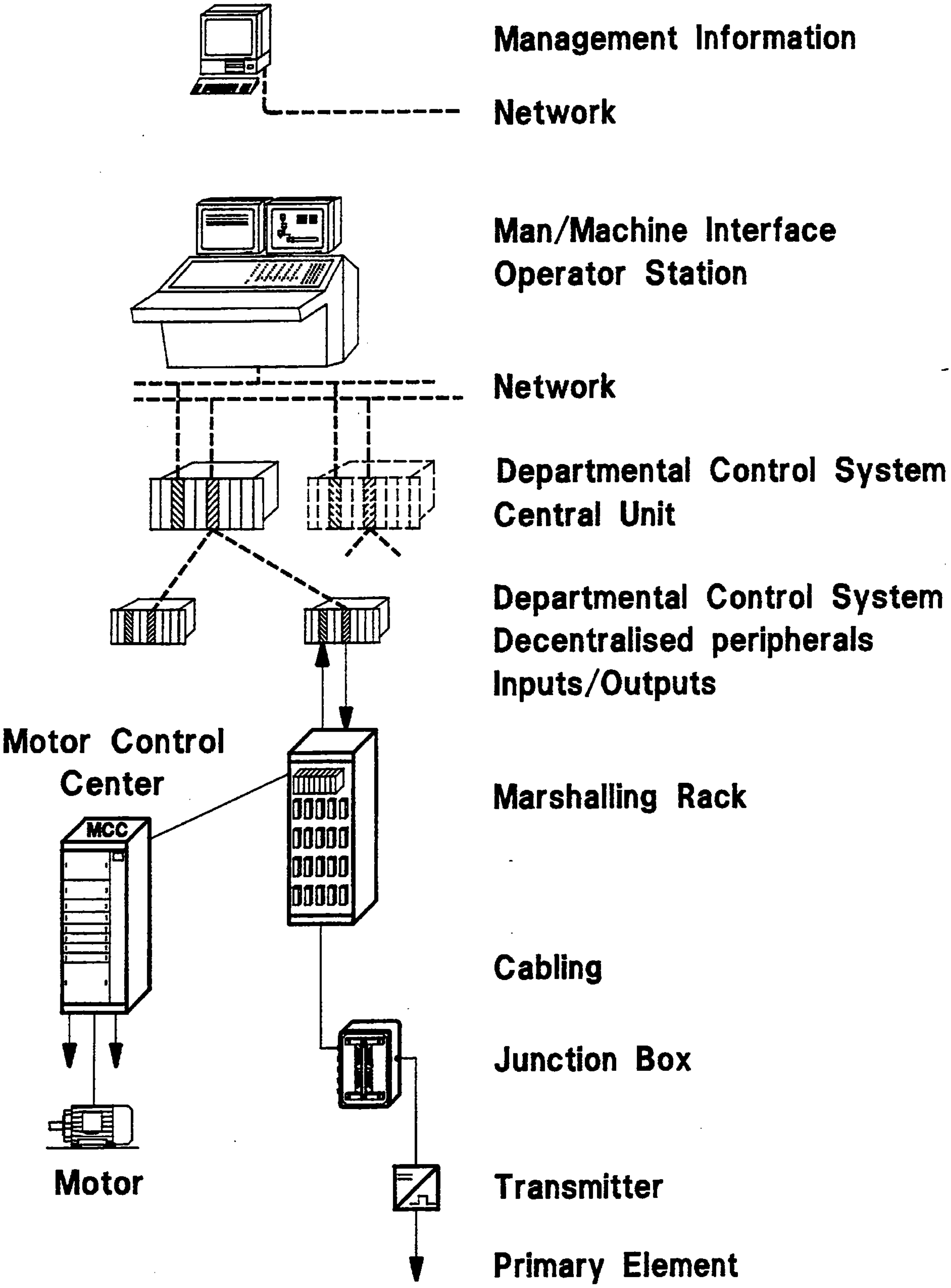
1 cycle = 10...300 msec
PROGRAM LENGTH = 1000...100,000 INSTRUCTIONS

Understanding Computer Technology

Understanding Computer Technology

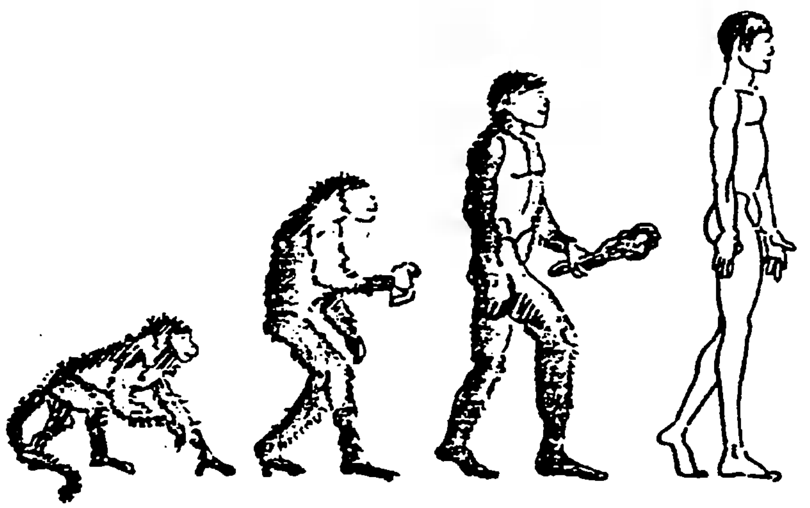


Automation systems - Typical configuration



History of PLC's

1968:	Concept of a Programmable Controller
1969:	Hardware based CPU Logic statements, 1 Kbytes Memory, 128 i/o's
1973:	Source Code Editing
1974:	PLC with Multi-Processors Logic, Counter, Timer, Move Words, Arithmetic's 12 Kbytes Memory, 1024 i/o's
1976:	Decentralised I/O-Systems
1977:	Micro Processor based PLC with Logic Co-Processors
1978:	Universal I/O-Structure
1979:	Bit Slice Processor Architecture
1980:	High efficient decentralised I/O-Structure with intelligent I/O-Modules and Block Transfer Instructions
1981:	Data Highway with Medium Response Time(Token Passing)
1982/83:	Micro-programmed, with multi-processor PLC's of 4th Generation In BASIC programmable Co-Processors Integrated Hard disk
1985:	Intelligent Programming Units Optical Links Use of PC's for programming and debugging
1986	Graphic/Function Programming More Mass-Storage
1989:	?



1968 **M**odular **D**igital **C**ontroller

MODICON

- General Motors:
  - simple to program, easy to change
  - maintenance low → modular system
  - safe in industrial environment
  - size smaller than relay panels
  - data communication to other systems
  - cheaper than relais

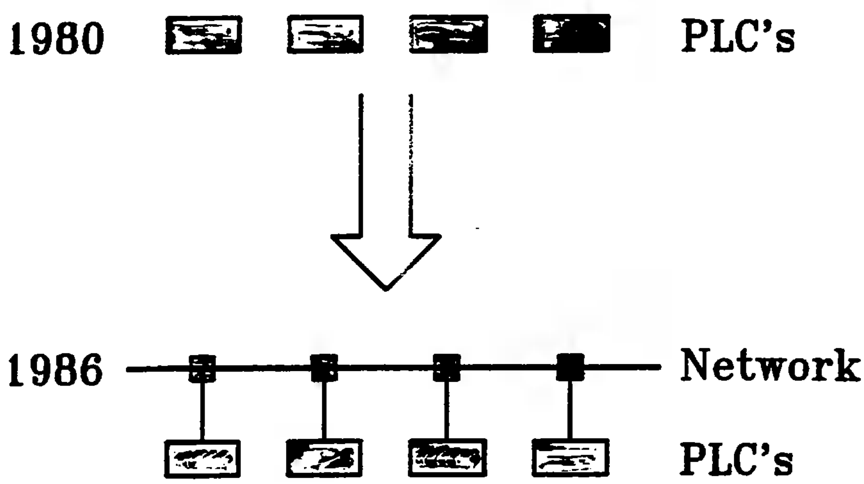
⇓

Programmable Logic Controller

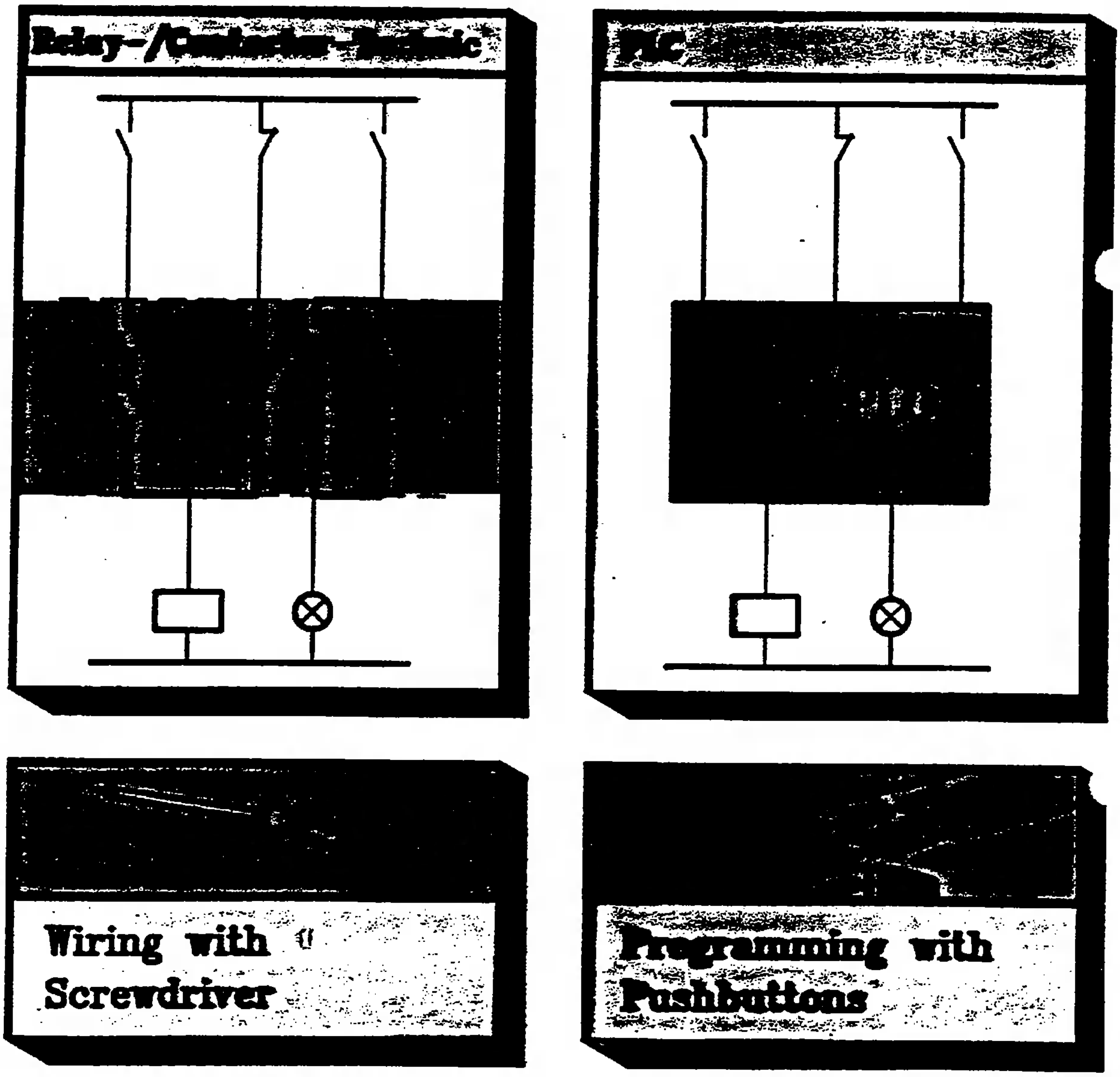
Evolution of Control Systems

40 years of Relay Systems
20 years of Solid State
10 years of PLC's

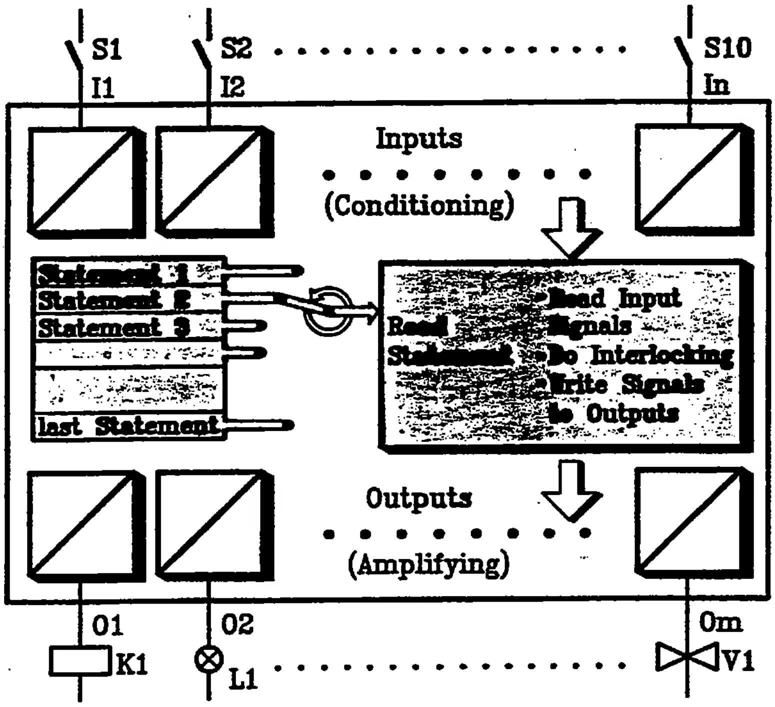
from island to system



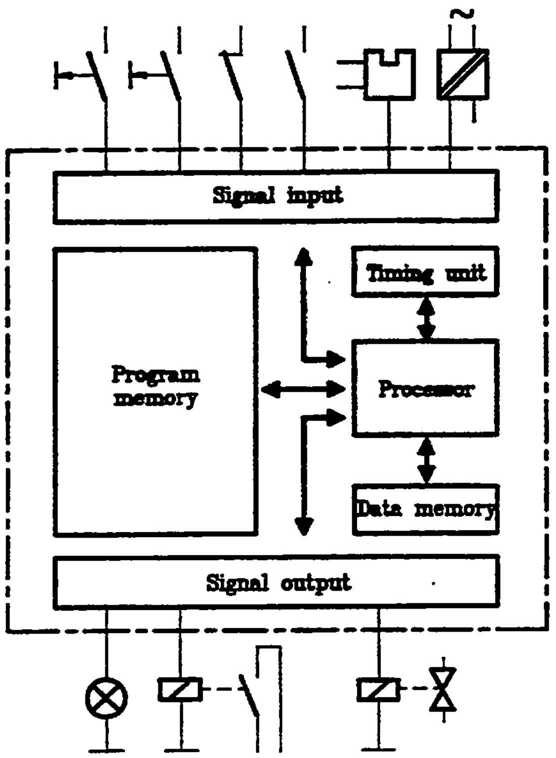
Differences between Relay and PLC Programming



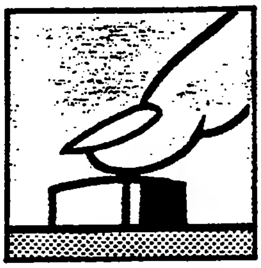
Working Principle of a PLC



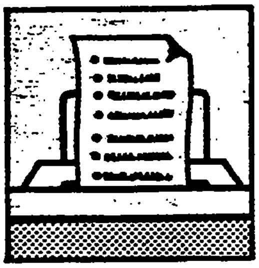
Block diagram of programmable controller



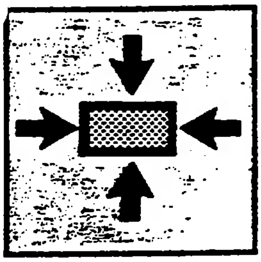
Advantages of PLC control against relay



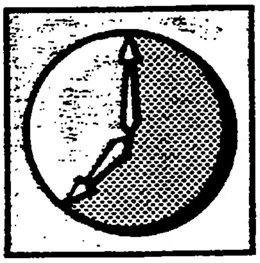
Change of control function by touching buttons



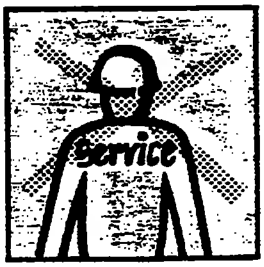
Automated and sophisticated documentation



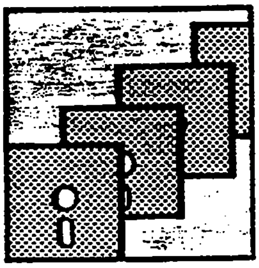
Small in size



Time saving while project execution



No wearing parts therefore maintenance free



Copy of programs by touching a button

Easy and fast for fault diagnostic

PLC-Progamming Language